Effective and Stable Role-Based Multi-Agent Collaboration by Structural Information Principles

Xianghua Zeng¹, Hao Peng¹, Angsheng Li¹,²

¹ State Key Laboratory of Software Development Environment, Beihang University, Beijing, China; ² Zhongguancun Laboratory, Beijing, China.
{zengxianghua, penghao, angsheng}@buaa.edu.cn, liangsheng@gmail.zgclab.edu.cn.

Abstract
Role-based learning is a promising approach to improving the performance of Multi-Agent Reinforcement Learning (MARL). Nevertheless, without manual assistance, current role-based methods cannot guarantee stably discovering a set of roles to effectively decompose a complex task, as they assume either a predefined role structure or practical experience for selecting hyperparameters. In this article, we propose a mathematical Structural Information principles-based Role Discovery method, namely SIRD, and then present a SIRD optimizing MARL framework, namely SR-MARL, for multi-agent collaboration. The SIRD transforms role discovery into a hierarchical action space clustering. Specifically, the SIRD consists of structuralization, sparsification, and optimization modules, where an optimal encoding tree is generated to perform abstracting to discover roles. The SIRD is agnostic to specific MARL algorithms and flexibly integrated with various value function factorization approaches. Empirical evaluations on the StarCraft II micromanagement benchmark demonstrate that, compared with state-of-the-art MARL algorithms, the SR-MARL framework improves the average test win rate by 0.17%, 6.08%, and 3.24%, and reduces the deviation by 16.67%, 30.80%, and 66.30%, under easy, hard, and super hard scenarios.

Introduction
Cooperative multi-agent reinforcement learning (MARL) has broadly been applied to a variety of complex decisions, such as games (Nowé, Vranx, and Hauwere 2012; Vinyals et al. 2019), sensor networks (Zhang and Lesser 2011), social network (Peng et al. 2022), emergent tool usage (Baker et al. 2020), etc. There are two significant challenges in cooperative MARL: scalability that the joint state-action space grows exponentially with the number of agents (Yang and Wang 2020), and partial observability, which necessitates decentralized policies based on local action-observation histories because of communication constraints (Nguyen, Nguyen, and Nahavandi 2020). Hence, the paradigm of Centralized Training with Decentralized Execution (CTDE) (Oliehoek, Spaan, and Vlassis 2008; Kraemer and Banerjee 2016; Mahajan et al. 2019) is designed to deal with the above challenges. However, the centralized training requires searching agent policies in the joint state-action space, making the CTDE challenging to learn efficiently, especially when the number of agents is large (Samvelyan et al. 2019). A valid solution is integrating roles to decompose tasks in multi-agent systems, where each role is associated with a particular subtask and a role policy in the restricted state-action space (DeLoach and García-Ojeda 2010; Bonjean et al. 2014; Sun, Liu, and Dong 2020).

The key is bringing forward a set of roles to decompose the cooperative task effectively. The typical methods to predefine task decomposition or roles (Lhaksmana, Mukrami, and Ishida 2018; Sun, Liu, and Dong 2020) require prior knowledge (subtask-specific rewards or role responsibilities) that is unavailable in practice. It is impractical to automatically learn an appropriate set of roles from scratch, equivalent to learning in the joint state-action space with substantial explorations (Wilson, Fern, and Tadepalli 2010; Wang et al. 2020). Instead of learning roles from scratch, RODE (Wang et al. 2021b) proposed to use the clustering technology to achieve role discovery in the joint action space. However, the performance of the RODE depends heavily on practical experience due to its high sensitivity to parameters of adopted clustering algorithms, such as the cluster number of K-means, the maximum and minimum cluster number of X-means, and the neighborhood radius and density threshold of DBSCAN. Because practical task decomposition is not always well-defined a priori or changing, current role-based methods (Sun, Liu, and Dong 2020; Wang et al. 2020, 2021b) cannot guarantee effective role discovery without manual assistance.

This paper proposes a mathematical Structural Information principles-based Role Discovery method, namely SIRD, and then presents a SIRD optimizing MARL framework, namely SR-MARL, for multi-agent collaboration. The crucial insight is that, instead of flat clustering, the SIRD transforms role discovery into hierarchical action space clustering. We firstly construct a weighted, undirected, and complete action graph, whose vertices represent actions, edges represent action correlations, and edge weight quantifies the functional correlation between actions with respect to achieving team targets. Secondly, although recently proposed methods (Karami and Johansson 2014; Zhang et al. 2022), achieving automatical parameters search, introduces significant computational overhead, which is not conducive to real-time decision-making.
we leverage the one-dimensional structural entropy minimization principle (Li, Yin, and Pan 2016) to sparsify the action graph, to generate an initial encoding tree. Thirdly, we minimize the $K$-dimensional structural entropy (Li et al. 2018) of the sparse graph to get the optimal encoding tree, thereby achieving the hierarchical action space clustering. Furthermore, we take the hierarchical clustering on the optimal encoding tree as hierarchical abstracting on the encoding tree guided by one-dimensional and role discovery into a hierarchical abstracting on the encoding tree, thereby achieving the hierarchical action space clustering. The SIRD is independent of manual assistance and flexibly integrated with various value function factorization approaches. Extensive experiments are conducted on the StarCraft II micromanagement tasks, including five easy maps, four hard maps, and four super hard maps. Comparative results and analysis demonstrate the performance advantages of our proposed role discovery method. All source code and data are available at Github\footnote{https://github.com/RingBDStack/SR-MARL}.

The main contributions of this paper are as follows: 1) To the best of our knowledge, it is the first time to incorporate probability function, clustering, the hierarchical clustering achieved by the encoding tree guarantees more effective role discovery without manual assistance. 2) An innovative method, which transforms coding tree guarantees more effective role discovery without manual assistance. 4) The extraordinary performance on challenging tasks shows that the SR-MARL achieves up to 6.08\% average test win rate improvement and up to 66.30\% deviation reduction than state-of-the-art baselines.

Preliminaries

Dec-POMDP

In this work, we consider a fully cooperative multi-agent task $M$, which can be modeled as a Dec-POMDP (Oliehoek and Amato 2016). The Dec-POMDP is defined as a tuple $(N, S, A, P, \Omega, O, R, \gamma)$, where $N \equiv \{n_1, n_2, \ldots, n_{|N|}\}$ is the finite set of agents, $S$ is the finite set of global states, $A$ is the action space, $P$ is the global state transition function, $\Omega$ is the finite set of partial observations, $O$ is the observation probability function, $R$ is the joint reward function, and $\gamma \in [0,1)$ is the discount factor. At each timestep, each agent $n_i \in N$ chooses an action $a_i \in A$ on a global state $s \in S$, and all chosen actions form a joint action $a \equiv [a_i]_{i=1}^{|N|}$. The joint action $a$ results in a joint reward $r = R(s, a)$ and a transition to the next global state $s' \sim P(\cdot | s, a)$. We consider the partially observable setting, where each agent $n_i$ receives an individual partial observation $o_i \in \Omega$ according to the function $O(o_i | s, a_i)$. Each agent $n_i$ has an action-observation history $\tau_i \in T$ and constructs individual policy $\pi_i(a_i | \tau_i)$ to maximize team performance jointly.

Role-based Learning

The idea of role-based learning is to decompose a multi-agent cooperative task into a set of subtasks, where the role set specifies task decomposition and subtask policies (Wilson, Fern, and Tadepalli 2010; Wang et al. 2021b). Given a fully cooperative multi-agent task $M$, let $\Psi$ be the role set. Each role $\rho_j \in \Psi$ is defined as a tuple $\langle \tau_j, \pi_{\rho_j} \rangle$, where $\tau_j$ is a subtask defined as $\langle N_j, S, A_j, P, \Omega, O, R, \gamma \rangle$, $N_j \subset N, \cup_j N_j = N$, and $N_j \cap N_l \neq \emptyset, j \neq l$. $A_j$ is the restricted action space of role $\rho_j$, $A_j \subset A, \cup_j A_j = A$, and $|A_j \cap A_l| \geq 0, j \neq l$. $\pi_{\rho_j} : T \times A_j \rightarrow [0,1]$ is the role policy for the subtask $\tau_j$.

Structural Information Principles

In the structural information principles (Li and Pan 2016), the structural entropy measures the uncertainty of a graph under a strategy of hierarchical partitioning. And the dynamics measurement of the encoding tree is recently analyzed (Yang, Peng, and Li 2022). By minimizing the $K$-dimensional structural entropy, the optimal hierarchical structure of the graph is generated, namely the optimal partitioning tree, which we call an “encoding tree” in our article. Given a weighted undirected graph $G = (V, E, W)$, $V$ is the vertex set, $E$ is the edge set, and $W : E \mapsto \mathbb{R}^+$ is the weight function. Let $n = |V|$ be the number of vertices and $m = |E|$ be the number of edges. For each vertex $v \in V$, its degree $d_v$ is defined as the sum of the weights of its connected edges. Then, we formally give the definitions of the encoding tree, the one-dimensional, and $K$-dimensional structural entropy of the weighted undirected graph $G$.

Encoding Tree. The encoding tree of $G$ is defined as a rooted tree $T$ with the following properties: 1) For each node $\alpha \in T$, there is a vertex subset in $G$ corresponding with $\alpha$, $T_\alpha \subseteq V$. 2) For the root node $\lambda$, we set $T_\lambda = V$. 3) For each node $\alpha \in T$, its children nodes are marked as $\alpha^\wedge(i)$ ordered from left to right as $i$ increases, and $\alpha^\wedge(0)^- = \alpha$. 4) For each node $\alpha \in T$, we suppose that $L$ is the number of its children nodes; then all vertex subsets $T_{\alpha^\wedge(i)}$ are disjointed, and $T_\alpha = \bigsqcup_{i=1}^L T_{\alpha^\wedge(i)}$. 5) For each leaf node $\nu$, $T_\nu$ is a singleton containing a single vertex in $V$.

One-dimensional Structural Entropy. The one-dimensional structural entropy of $G$ is defined as follows:

$$H^1(G) = - \sum_{v \in V} \frac{d_v}{\text{vol}(G)} \cdot \log_2 \frac{d_v}{\text{vol}(G)},$$

where $\text{vol}(G) = \sum_{v \in V} d_v$ is the volume of $G$.

$K$-dimensional Structural Entropy. Given an encoding tree $T$ whose height is at most $K$, the $K$-dimensional structural entropy of $G$ is defined as follows:

$$H^K(G) = \min_T \left\{ \sum_{\alpha \in T, \alpha \neq \lambda} H^T(G; \alpha) \right\},$$

$$H^T(G; \alpha) = - \frac{g_\alpha}{\text{vol}(G)} \log_2 \frac{\nu_\alpha}{\nu_\alpha - V_\alpha},$$

where $g_\alpha$ is the sum of weights of all edges connecting vertices in $T_\alpha$ with vertices outside $T_\alpha$, $V_\alpha$ is the volume of $T_\alpha$.
the sum of degrees of all vertices in $T_\alpha$, and $T$ ranges over all encoding trees of height at most $K$. The dimension $K$ is also the maximal height of the encoding tree $T$.

**The SIRD Optimizing MARL Framework**

In this section, we present the overall framework of the SR-MARL and describe the detailed design of the structural information principles-based role discovery method SIRD.

Overall Framework of SR-MARL

The SR-MARL consists of five modules: Environment, Agent Network $Q_i$, Mixing Network $Q_{tot}$, Role Selector, and the role discovery module SIRD, as shown in Fig.1.

In the overall framework, each agent $n_i$ makes decisions based on individual network $Q_i$, which takes the partial observation $o_i$ and joint reward $r$ as inputs and is updated by the QPLEX-style mixing network $Q_{tot}$ (Wang et al. 2021a). The mixing network $Q_{tot}$ has access to global information for centralized training. Apart from making decisions, the individual network $Q_i$ encodes the local action-observation history $\tau_i$ into a $d$-dimensional hidden vector $h_{\tau_i}$, which is then fed into the role selector. The role discovery module SIRD described in the next subsection takes the action-observation histories of all agents $[\tau_i]_{i=1}^N$ and joint reward $r$ as inputs. And the SIRD outputs action representations $z_{a_i}$, role representations $z_p$ and restricted action spaces $A_p$ to the role selector for learning role policies. Inspired by the RODE (Wang et al. 2021b), the role selector assigns role $\rho_j \in \Psi$ and its associated action subspace $A_j \subset A_p$ to agent $n_i$, based on the dot product between the role representations $z_p$ and the hidden vector $h_{\tau_i}$. For coordinating the role assignment of all agents, we additionally apply a duplex dueling network of QPLEX (Wang et al. 2021a) to mix the dot products.

Role Discovery Module SIRD

As shown in Fig.1, the SIRD comprises Structuralization, Sparsification, and Optimization. In the structuralization, we map the action space to a fixed-dimensional embedding space and construct an action graph. In the sparsification, we sparsify the action graph and generate an initial encoding tree of the sparse graph. In the optimization, we optimize the encoding tree to realize the optimal hierarchical clustering, equivalent to a hierarchical abstracting of actions, and achieve role discovery on the optimal encoding tree.

**Structuralization.** Unlike the existing role-based methods (Sun, Liu, and Dong 2020; Wang et al. 2020, 2021b), the SIRD utilizes action correlations to construct an action graph, to improve the effectiveness of the role discovery. To this end, we learn action representations for achieving team targets, measure correlations between representations, and then construct the weighted, undirected, complete action graph based on the correlations.

Firstly, inspired by the RODE (Wang et al. 2021b), we similarly adopt the encoder-decoder structure (Cho et al. 2014) to learn the action representations, mapping the action space $A$ to a $d$-dimensional embedding space. In the encoder, we encode each action $a \in A$ as an embedded representation $z_a \in \mathbb{R}^d$, as the step 1 in Fig.2. For each agent $n_i$, the decoder decodes the representation $z_{a_i}$ of its adopted action $a_i$ to reconstruct its partial observation $\tau_i' = \tau_i$ and joint reward $r$. Given the action-observation histories of all agents $[\tau_i]_{i=1}^N$ and joint reward $r$, the encoder-decoder structure is trained end-to-end by minimizing the reconstruction loss. Secondly, for every two actions $a_i$ and $a_j$ with $a_i \neq a_j$, we measure their correlation $C_{a_i,a_j} \in [-1, 1]$ through the Pearson Correlation Analysis:

$$C_{a_i,a_j} = \frac{E((z_{a_i} - \mu_{z_{a_i}})(z_{a_j} - \mu_{z_{a_j}}))}{\sigma_{z_{a_i}} \sigma_{z_{a_j}}} .$$

where $\mu_{z_{a_i}}$ and $\sigma_{z_{a_i}}$ are the mean value and variance of the $d$-dimensional action representation $z_{a_i}$, respectively. Intuitively, the larger absolute value of $C_{a_i,a_j}$ represents a more functional correlation between action $a_i$ and action $a_j$ on achieving team targets. Thirdly, we take each action in $A$ as a vertex, connect any two vertices $a_i$ and $a_j$ and assign $C_{a_i,a_j}$ to edge $(a_i, a_j)$ as weight $w_{ij} = C_{a_i,a_j}$, to construct the action graph $G$, as the step 2 in Fig.2.

Therefore, in the action graph $G$, vertices represent actions in the action space $A$, $V = A$, edges represent action correlations, and edge weight further quantifies the functional correlation between actions.

**Sparsification.** We implement sparsification of the action graph to reduce the computational cost and eliminate negative interference of trivial weights whose absolute values approach 0. Following the construction of cancer cell neighbor networks (Li, Yin, and Pan 2016), we sparsify the action graph $G$ into a $k$-nearest neighbor ($k$-NN) graph $G_k$ based on the one-dimensional structural entropy minimization principle. By minimizing the one-dimensional structural entropy of $G_k$, $H^1(G_k)$, we select the optimal number of neighbors $k^*$, as the step 3 in Fig.2. We summarize the sparsification step as Alg.1.

![Figure 1: The overall framework of the SR-MARL.](image-url)
Firstly, based on the mean value $\mu_w$ of edge weights in $G$, we introduce a factor $\delta, \delta = \frac{1}{|A|} \cdot \mu_w, |A|$ is the number of actions, to correct all weights in $G$ (Lines 2-4 in Alg.1):

$$w_{ij} = w_{ij} + \delta. \quad (5)$$

For significant weights whose absolute values approach 1, the factor $\delta$ is relatively tiny, and the corrected weights are approximately equal to the original values. For trivial weights, $\delta$ can notably correct them to eliminate their negative interference. Secondly, for each $k$, we construct the graph $G_k$ (Lines 5 and 8 in Alg.1) by solely retaining the most significant $k$ edge weights for each vertex and then calculate the one-dimensional structural entropy $H^1(G_k)$ (Lines 6 and 9 in Alg.1). Thirdly, we find all local minimal structural entropies (Lines 10-11 in Alg.1), namely LMSE, select the minimum $k^*$ from the LMSE, and output $G_{k^*}$ as the sparse action graph $G^*$ (Lines 12-14 in Alg.1).

Therefore, we generate an initial encoding tree $T$ of $G^*$: 1) For the action space $A$, we generate a root node $\lambda$ and set its corresponding vertex subset $T_\lambda = A$; 2) For each action $a \in A$, we generate a leaf node $\nu$ with $T_\nu = \{a\}$ and set $\nu$ as a child node of $\lambda, \nu = \lambda$, as the step 4 in Fig.2. Intuitively, the height of the initial encoding tree $T$ is 1. The associated hierarchical clustering is initialized as a single-level hierarchy where each action is divided into a separate category, as the initial clustering $C$ in Fig.2.

**Optimization.** To realize the optimal hierarchical clustering $C^*$ of the action space $A$, we optimize the encoding tree $T$ of the sparse action graph $G^*$ from 1 layer to $K$ layers. Firstly, we introduce two operators $merge$ and $combine$ from the deDoc (Li et al. 2018) to minimize the $K$-dimensional structural entropy of the sparse graph $G^*$ for optimizing $T$, as the step 5 in Fig.2. In the encoding tree $T$, two nodes are brothers if they have a common father node. The $merge$ and $combine$ operators are defined on brother nodes and marked as $T_{mg}$ and $T_{cb}$ in our work. Secondly, based on the $K$-dimensional structural entropy minimization principle (Li et al. 2018), we utilize $T_{mg}$ and $T_{cb}$ to design an iterative optimization algorithm, Alg.2. At each iteration, Alg.2 traverses all brother nodes $\beta_1$ and $\beta_2$ in $T$ (Lines 4 and 9 in Alg.2) and greedily selects operator $T_{mg}$ or $T_{cb}$ that reduces the structural entropy the most (Lines 5 and 10 in Alg.2) to execute (Lines 7 and 12 in Alg.2) on the premise that the tree height does not exceed $K$. When no brother nodes satisfy $\Delta SE > 0$, we terminate the iteration and output $T$ as the optimal encoding tree $T^*$. In the encoding tree $T^*$, the root node $\lambda$ corresponds to the action space $A, T_\lambda = A$, each leaf node $\nu$ corresponds to a singleton containing a single action $a \in A$, and other nodes correspond to clusters of different hierarchies. For each leaf node $\nu$ with $T_\nu = \{a\}$, we set its node representation $z_\nu = z_a$. Thirdly, we realize the optimal hierarchical action space clustering $C^*$ according to $T^*$.

![Figure 2: The structural information principles-based role discovery.](image_url)
the optimal encoding tree $T^*$ as hierarchical abstracting of actions to discover roles, as the step 6 in Fig.2.

**Role Discovery on the Optimal Encoding Tree.** The optimal encoding tree $T^*$ partitions the action space $A$ into a 3-level abstracting hierarchy from up to bottom, including roles, sub-roles, and actions, as shown in Fig.3(a). In the 3-level hierarchy, children nodes of the root node are defined as roles $\Psi \equiv [\rho_i]^{\Psi}, \rho_i = \lambda^\Psi(i)$, leaf nodes $[\nu_i]^{A}$ are defined as actions, and other nodes on the paths connecting roles and actions are defined as sub-roles $[\rho'_j]$ as shown in Fig.3(b). For example, actions $\nu_{10}, \nu_{11}, \nu_{13}$ are abstracted as a sub-role $\rho'_2$, and then both $\rho'_2$ and action $\nu_{12}$ are abstracted as a role $\rho_4$. To calculate role representation $z_{\rho^i}$, we need to hierarchically aggregate node representations from bottom to up in the subtree rooted by $\rho_i$. For example, we aggregate action representations $z_{\nu_{10}}, z_{\nu_{11}}, \nu_{13}$ and $z_{\nu_{12}}$ as a sub-role representation $z_{\rho'_2}$, and then aggregate $z_{\rho'_2}$ and $z_{\nu_{12}}$ as a role representation $z_{\rho_4}$.

To realize aggregating from bottom to up, we propose using the structural entropy distribution as each node’s weight. For each non-leaf node $\alpha \in T^*$, the aggregate function is defined to calculate its node representation $z_{\alpha}$ as follows:

$$z_{\alpha} = \sum_{i=1}^{L} \frac{H^{T^*}(G; \alpha^\Psi(i))}{\sum_{j=1}^{L} H^{T^*}(G; \alpha^\Psi(j))} \cdot z_{\alpha^\Psi(i)}, \quad (6)$$

where $L$ is the number of children nodes of $\alpha$. In addition, the restricted action space of role $\rho_i$ is defined as its corresponding vertex subset $T_{\rho_i}, A_1 = T_{\rho_i}$.

Finally, we output action representations $z_{\alpha} \equiv \{z_{\alpha} \mid \forall \alpha \in A\}$, role representations $z_{\rho} \equiv \{z_{\rho} \mid \forall \rho \in \Psi\}$, and restricted action spaces $A_\rho \equiv \{A_1 \mid \forall \rho \in \Psi\}$ to achieve the role discovery.

**Time Complexity of SIRD.** The overall time complexity of the SIRD is $O(n^2 + n + n \cdot \log^2 n)$, where $n$ is the number of actions, $n = |A|$. Here, the time complexity of the structuralization is $O(n^2)$, which analyzes Pearson Correlation for every two actions. In the sparsification, the SIRD costs $O(n)$ time complexity to find the minimum $k^\star \in \{1, \ldots, n\}$ from all local minimal structural entropies. In our work, the optimization of the encoding tree costs $O(n \cdot \log^2 n)$ time complexity by adopting a similar data structure (Clauset, Newman, and Moore 2004). Compared with flat clustering in the RODE (Wang et al. 2021b), the SIRD achieves role discovery in the same quadratic time complexity and guarantees independency of manual assistance, which leads to effective and stable performance advantages.

**Experiments and Analysis**

In this section, we conduct a large set of experiments and comparative experiments, aiming to verify the effectiveness and stability of the SR-MARL. Towards fair evaluation, all results are illustrated with the average and deviation of the performance testing with different random seeds, like in other works (Wang et al. 2021a,b).

### Algorithm 2: The Iterative Optimization Algorithm

**Input**: The initial encoding tree $T$

**Output**: The optimal encoding tree $T^*$

```plaintext
1. $\Delta SE, \beta_1^*, \beta_2^* \leftarrow$ initialization
2. while True do
   3. $\Delta SE \leftarrow 0$
   4. for each brother nodes $\beta_1$ and $\beta_2$ in $T$ do
      5. $\Delta SE, \beta_1^*, \beta_2^* \leftarrow$ maximize the reduction of the structural entropy caused by the $merge$ operator via Eq.(2)
   6. if $\Delta SE > 0$ then
      7. $T \leftarrow T_{mb}(T; \beta_1^*, \beta_2^*)$
      8. Continue
   9. for each brother nodes $\beta_1$ and $\beta_2$ in $T$ do
      10. $\Delta SE, \beta_1^*, \beta_2^* \leftarrow$ maximize the reduction of the structural entropy caused by the $combine$ operator via Eq.(2)
   11. if $\Delta SE > 0$ then
      12. $T \leftarrow T_{cb}(T; \beta_1^*, \beta_2^*)$
   13. else
      14. Break
   15. $T^* \leftarrow T$
   16. return $T^*$
```

**Figure 3:** The role discovery on the optimal encoding tree.

**Experiment Setup**

**Datasets.** We evaluate the SR-MARL on the StarCraft II micromanagement (SMAC) benchmark (Samvelyan et al. 2019), a mainstream benchmark of CTDE algorithms, of its rich environment and high control complexity. The SMAC benchmark includes five easy maps, four hard maps, and four super hard maps, where hard and super hard maps are typically hard-exploration tasks requiring agents to learn complex collaboration. As the SR-MARL is presented for improving multi-agent collaboration, we primarily focus on its performance on hard and super hard maps. In the micromanagement scenarios, each unit is controlled by an independent agent acting based on local observation, and a built-in AI controls all enemy units. At each timestep, each agent selects an action from the discrete action space, which consists of moving in four directions, stopping, taking no-op, and selecting an enemy/ally unit to attack/heal.
Baselines and Variants. To make the empirical results more convincing, we compare the SR-MARL, whose maximal encoding tree height is set as 2, with state-of-the-art MARL algorithms, including independent Q-learning method (IQL (Tampuu et al. 2017)), value-based methods (VDN (Sunehag et al. 2018), QMIX (Rashid et al. 2018), QPLEX (Wang et al. 2021a), QTRAN (Son et al. 2019)), actor-critic method (COMA (Foerster et al. 2018)), and role-based method (RODE (Wang et al. 2021b)). The implementations of the SR-MARL and baselines in our experiments are based on the PyMARL ((Samvelyan et al. 2019)), and the hyperparameters of the baselines have been fine-tuned on the SMAC benchmark. All experiments adopt the default settings and are conducted on 3.00GHz Intel Core i9 CPU and NVIDIA RTX A6000 GPU.

Evaluations

We evaluate the SR-MARL and state-of-the-art MARL algorithms under different map categories (easy, hard, and super hard) and summarize averages and deviations of test win rates in Table 1. Table 1 shows that under different categories, our framework achieves at most a 6.08% improvement in average value and at most a 66.30% reduction in deviation. The improvement in average value and reduction in deviation correspond to the performance advantages on effectiveness and stability of the SR-MARL, respectively. In terms of effectiveness, explorations in restricted action spaces specified by the role set discovered from the optimal hierarchical abstracting guarantee strong cooperative ability among agents. In terms of stability, the SR-MARL leverages the structural information principles to guide automatic role discovery and therefore avoids the selection of sensitive hyperparameters. The performance advantages on effectiveness and stability of the SR-MARL become significant under hard-exploitation scenarios (hard and super hard maps).

On the other hand, we benchmark the SR-MARL and classical baselines on all 13 maps to evaluate the overall performance across the SMAC suite. Fig.4 shows each algorithm’s average test win rate across all maps and the number of maps where it achieves the highest average test win rate at different stages. Fig.4(left) illustrates that compared with baselines, the SR-MARL achieves outstanding overall performance and converges faster. In particular, the SR-MARL always maintains the highest average test win rate after 40% of the whole process until obtaining a 96.7% final average test win rate, which exceeds the second 90.99% (QPLEX) and the third 86.36% (RODE) by 5.71% and 10.34%. The advantages of the overall performance and learning efficiency can be attributed to the effective exploration of the action subspaces. Compared with the RODE, the SR-MARL accomplishes more effective role-based learning via transforming the role discovery into hierarchical action space clustering. From Fig.4(right), the SR-MARL finally performs best on almost half of all maps (5 of 13), much more than baselines.

In summary, the SR-MARL establishes a new state of the art on SMAC in terms of effectiveness and stability. Fig.5 show the learning curves of the SR-MARL and three representative baselines on each super hard map, respectively. The starting point of convergence and its variance are marked for each curve. The SR-MARL converges at 2056668 timestep and achieves a 94.6% average test win rate, with a variance of 0.0006, as shown in $27m_{vs30m}$.

Integrative Abilities. The SIRD is agnostic to specific MARL algorithms and can be integrated with various value function decomposition approaches by replacing the mixing network $Q_{tot}$ in Fig.1. We integrate the SIRD with the QMIX and QPLEX to get SR-QMIX and SR-QPLEX, respectively, and test their performance on map $2e_{vs64zg}$. All integrated frameworks outperform the original approaches in effectiveness and efficiency, as shown in Fig.6. The comparative results indicate that the structural information principles-based role discovery method can significantly enhance multi-agent coordination.

Ablation Studies. We design ablation studies on map $2e_{vs64zg}$ to evaluate the importance of structuralization and sparsification for performance advantages. To this end, we design two variants: ST-MARL and SP-MARL, degenerated frameworks of the SR-MARL without some functional modules. The ST-MARL variant without structuralization discovers roles from the joint action space via K-Means clustering instead of the SIRD module. The SP-MARL variant without sparsification directly optimizes the encoding tree of the complete action graph for role discovery. As
tiveness, stability, and efficiency, the SR-MARL-3 performs closely related to labor division and efficiency improvement. SR-MARL-3. Fig. 8 shows their learning curves on two different maps: (left) 2c_vs_64zg and (right) MMM2.

Figure 5: Average test win rates on four SMAC super hard maps.

Figure 6: Average test win rates of the SR-MARL integrated with value decomposition methods QMIX and QPLEX.

Figure 7: Average test win rates for ablation studies.

shown in Fig. 7, the SR-MARL significantly outperforms the ST-MARL in the average and the deviation of the test win rates, which shows structuralization is the foundation of the SIRD and is crucial for the performance advantages on effectiveness and stability. The comparison between SR-MARL and SP-MARL indicates that sparsification remarkably boosts the learning process without affecting the performance advantages.

We discuss different maximal tree heights (2 and 3) and separately name associated frameworks SR-MARL-2 and SR-MARL-3. Fig. 8 shows their learning curves on two different maps, 2c_vs_64zg and MMM2. In terms of effectiveness, stability, and efficiency, the SR-MARL-3 performs better on super hard map MMM2, while the SR-MARL-2 performs better on hard map 2c_vs_64zg. The reason is maybe that super hard maps require a more hierarchical action space abstracting associated with a higher encoding tree to achieve complex multi-agent collaboration.

**Conclusion**

This paper proposes a structural information principles-based role discovery method SIRD and presents a SIRD optimizing MARL framework SR-MARL for multi-agent collaboration. To the best of our knowledge, this is the first time that the mathematical structural information principles have been incorporated into MARL to improve performance under cooperative scenarios. Evaluations of challenging tasks in the SMAC benchmark demonstrate that the SR-MARL shows significant outperformance on effectiveness and stability over the state-of-the-art baselines and even can be flexibly integrated with various value function decomposition approaches to enhance coordination. For future work, we will conduct more analysis and explorations on the encoding tree under MARL scenarios.
Acknowledgments

The corresponding authors are Hao Peng and Angsheng Li. This paper was supported by the National Key R&D Program of China through grant 2021YFB1714800, NSFC through grant 61932002, S&T Program of Hebei through grant 20310101D, Natural Science Foundation of Beijing Municipality through grant 4222030, and the Fundamental Research Funds for the Central Universities.

References


